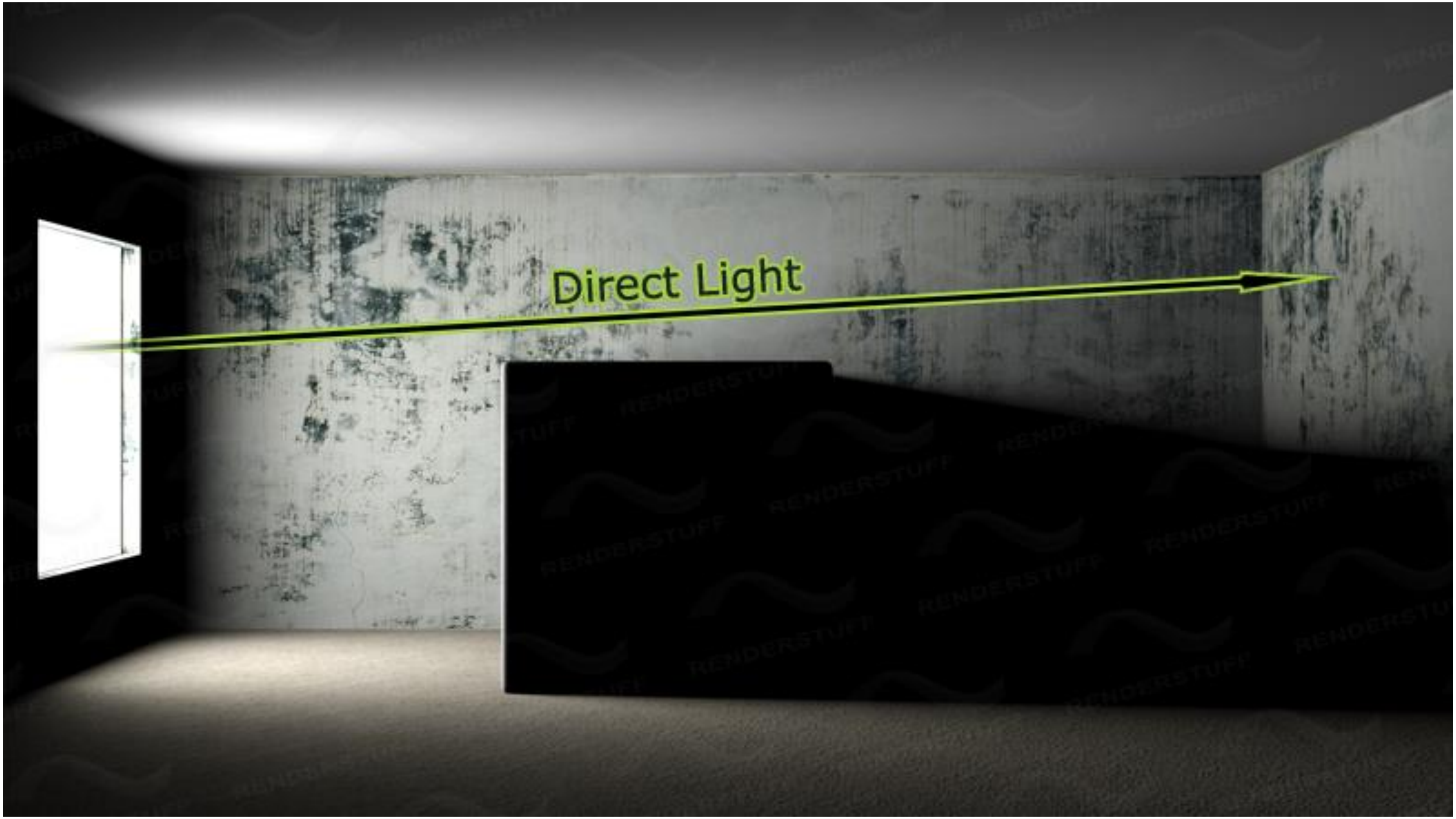
INTRODUCTION

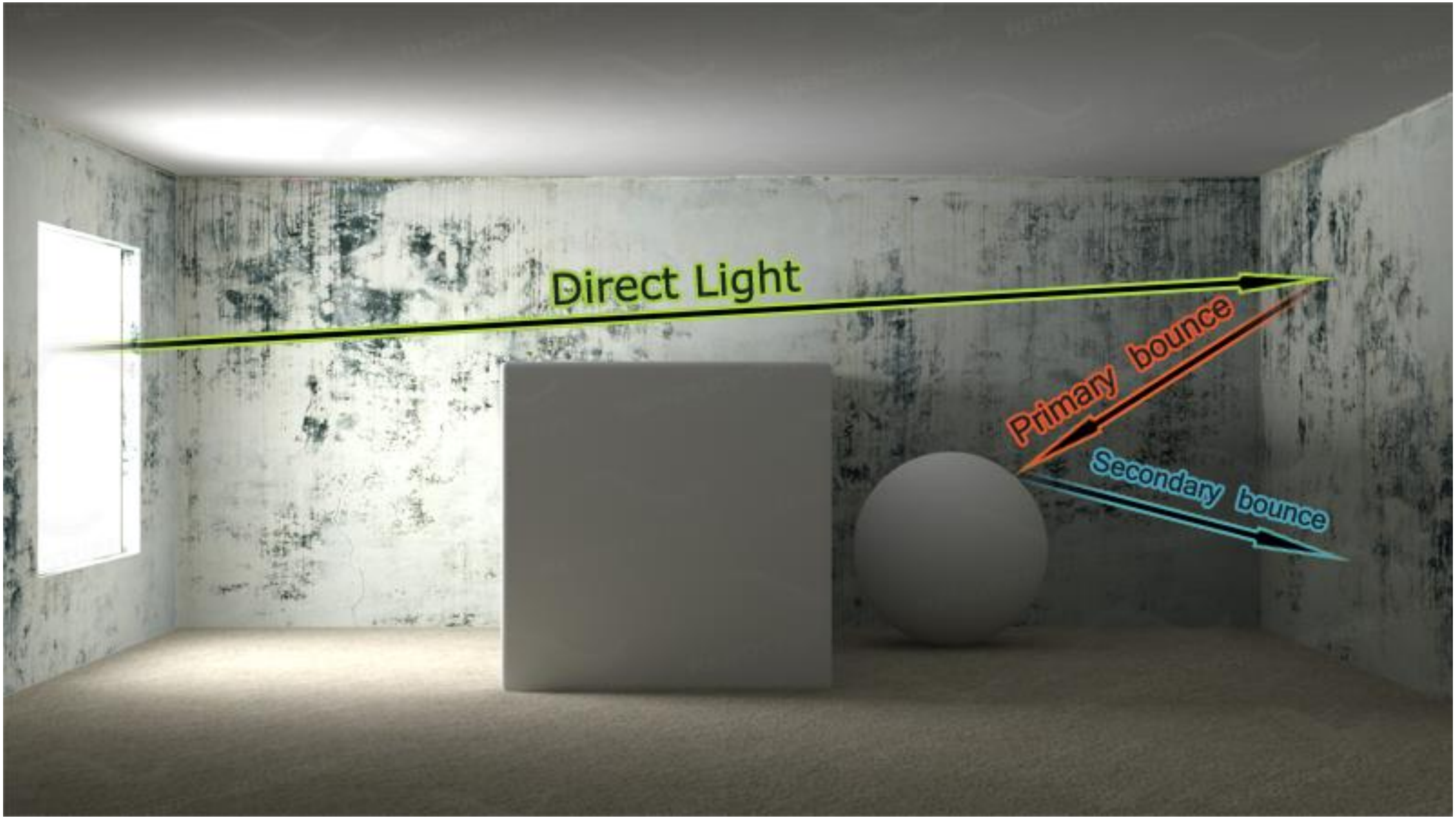
Motivation


Primary bounces
Multiplier: 1,0
Secondary bounces
Multiplier: 1,0

Motivation



Motivation



Motivation

Motivation

C# Introduction

- Garbage collector
  - return new Vector3(0, 0, 0);
- Namespaces
  - Math.Abs(x);
- Object oriented
  - int i = 1;
  - string s = i.ToString();

C# Objects

```
public class Vector3 {
    public static readonly Vector3 Zero = new Vector3(0, 0, 0);
    public Double X, Y, Z;
    public Double Length {
        get {
            return Math.Sqrt(this.X * this.X + this.Y * this.Y + this.Z * this.Z);
        }
    }
    public Vector3(Double x, Double y, Double z) {
        this.X = x;  this.Y = y;  this.Z = z;
    }
    public static Vector3 operator *(Vector3 a, Double b) {
        return new Vector3(b * a.X, b * a.Y, b * a.Z, 0);
    }
    public static Vector3 operator *(Double a, Vector3 b) {
        return new Vector3(a * b.X, a * b.Y, a * b.Z);
    }
}
```

C# Object Access

```
Vector3 P = new Vector3(0, 0, 0);
Vector3 Q = P – Vector3.Zero;
List<Vector3> list = new List<vector3>();
list.Add(P);
list.Add(Q);
foreach (Vector3 X in list) {
    Double length = X.Length;
}
```

RAY CASTING

Template 1

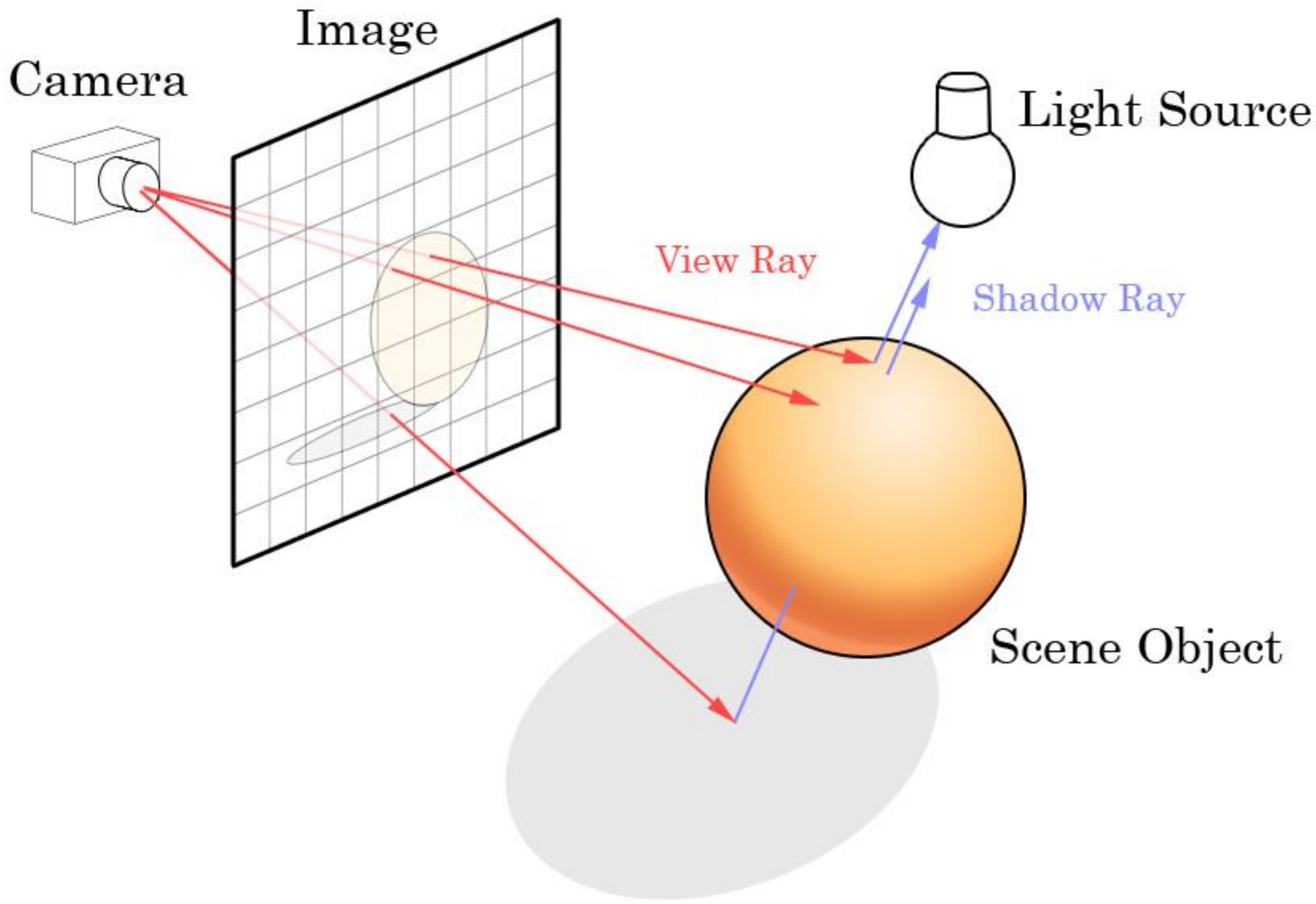
Computer Graphics 2

IDE & Vectors

- Visual Studio
- Sharp develop
- Mono develop, Xamarin Studio

- Basic vector operations are implemented
  - * serves as dot product
  - % serves as cross product

Ray Casting



Template

- Read camera parameters and render image
- Image is rendered by casting rays from camera through each pixel
- Pixel color is determined by ray intersection color

Ray

$$r(t) = P + t\boldsymbol{d}$$

- Ray r(t)
- Ray origin P
- Ray direction **d**
- Ray parameter t
- Ray hits an object if t >= 0

Plane

$$(X - Q) \cdot \boldsymbol{n} = 0$$

- X is arbitrary point
- Q is a point on the plane
- **n** is plane normal
- Ray-plane intersection needs to be calculated in order to determine pixel color

Ray – Plane Intersection

$$r(t) = P + t\boldsymbol{d}$$

$$(X - Q) \cdot \boldsymbol{n} = 0$$

$$(P + t\boldsymbol{d} - Q) \cdot \boldsymbol{n} = 0$$

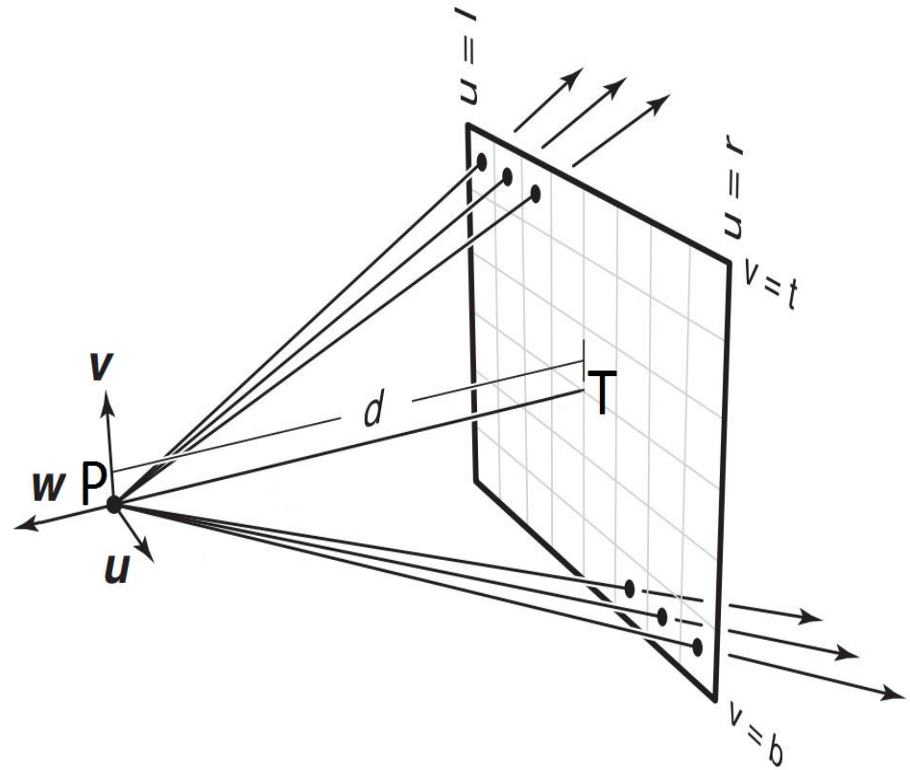
$$t\boldsymbol{d} \cdot \boldsymbol{n} = -(P - Q) \cdot \boldsymbol{n}$$

$$t\boldsymbol{d} \cdot \boldsymbol{n} = (Q - P) \cdot \boldsymbol{n}$$

$$t = \frac{(Q - P) \cdot \boldsymbol{n}}{\boldsymbol{d} \cdot \boldsymbol{n}}$$

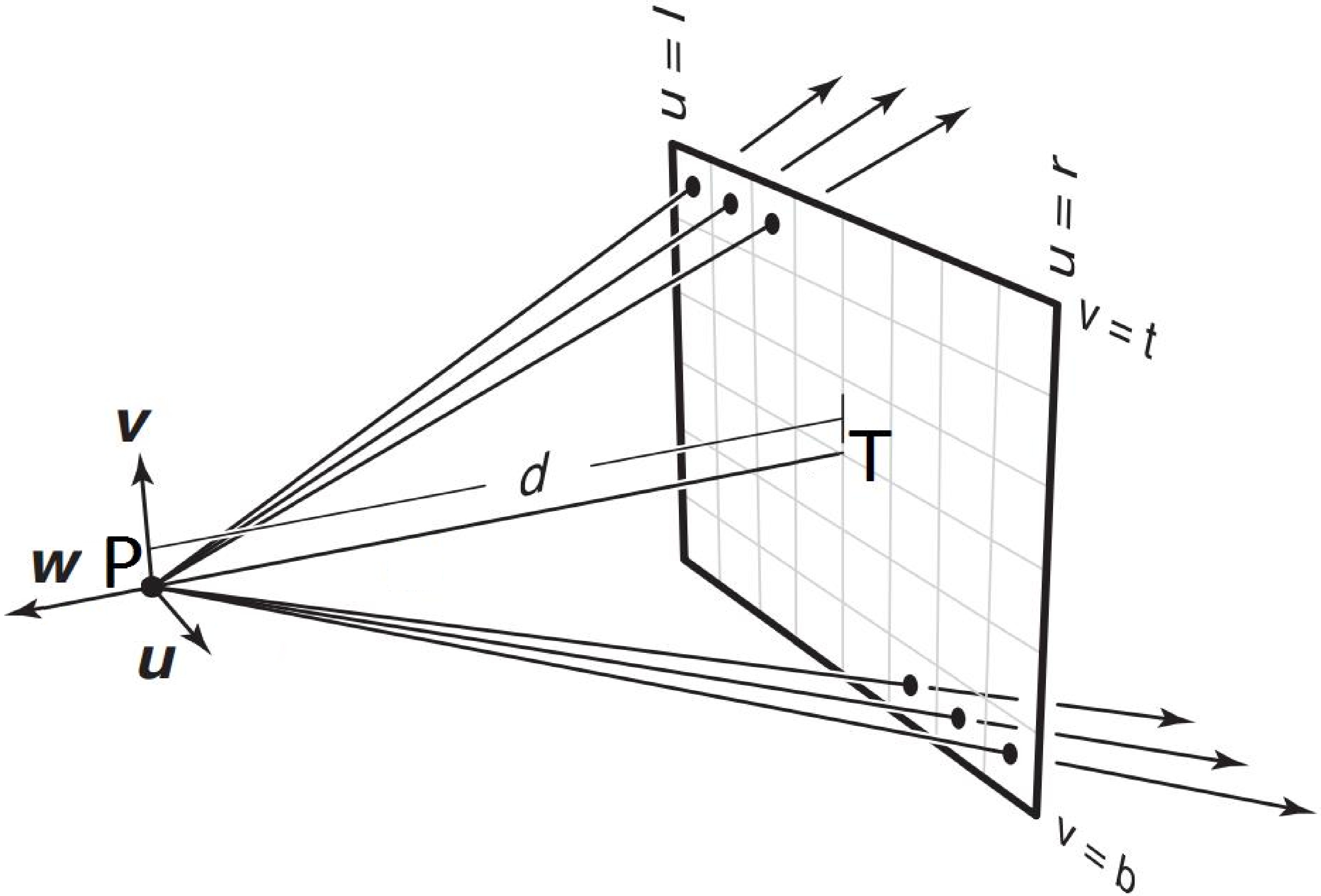
Camera

- P is position of camera
- Camera looks at target T
- **up** = (0, 0, 1)
- Look at direction of camera:
  w = (P - T)
- Camera right vector is:
  u = **up** × **w**
- Camera up vector:
  v = **w** × **u**
- Width and height determine screen size and aspect ratio
- Field of view Y determines visible space



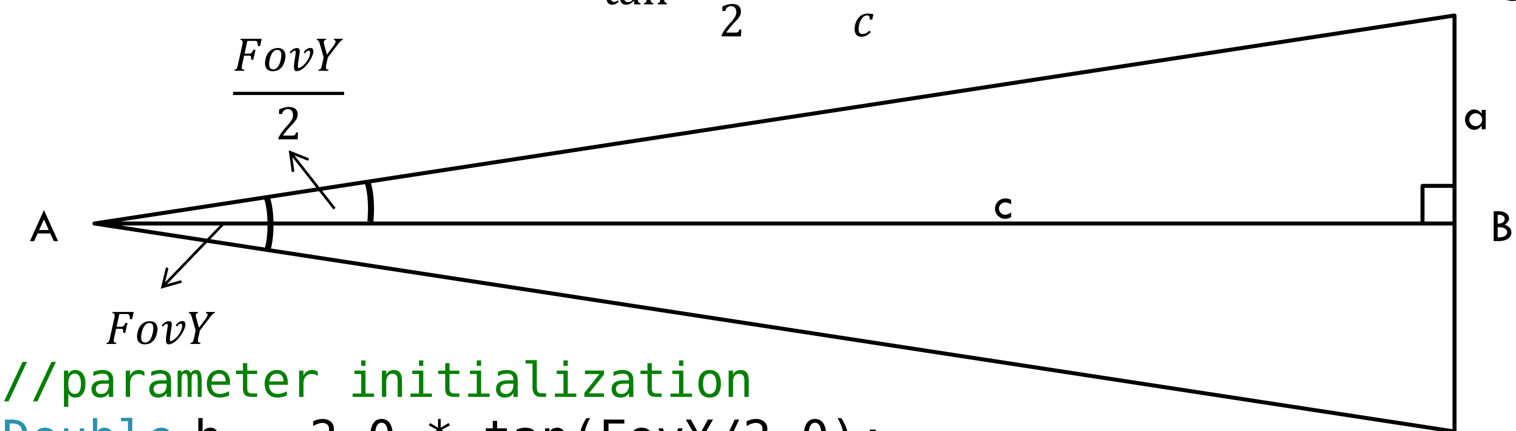
Camera

```
Vector3 dir = (u * U + v * V - W).Normalized;
Double aspect = (Double)Width / (Double)Height;
```

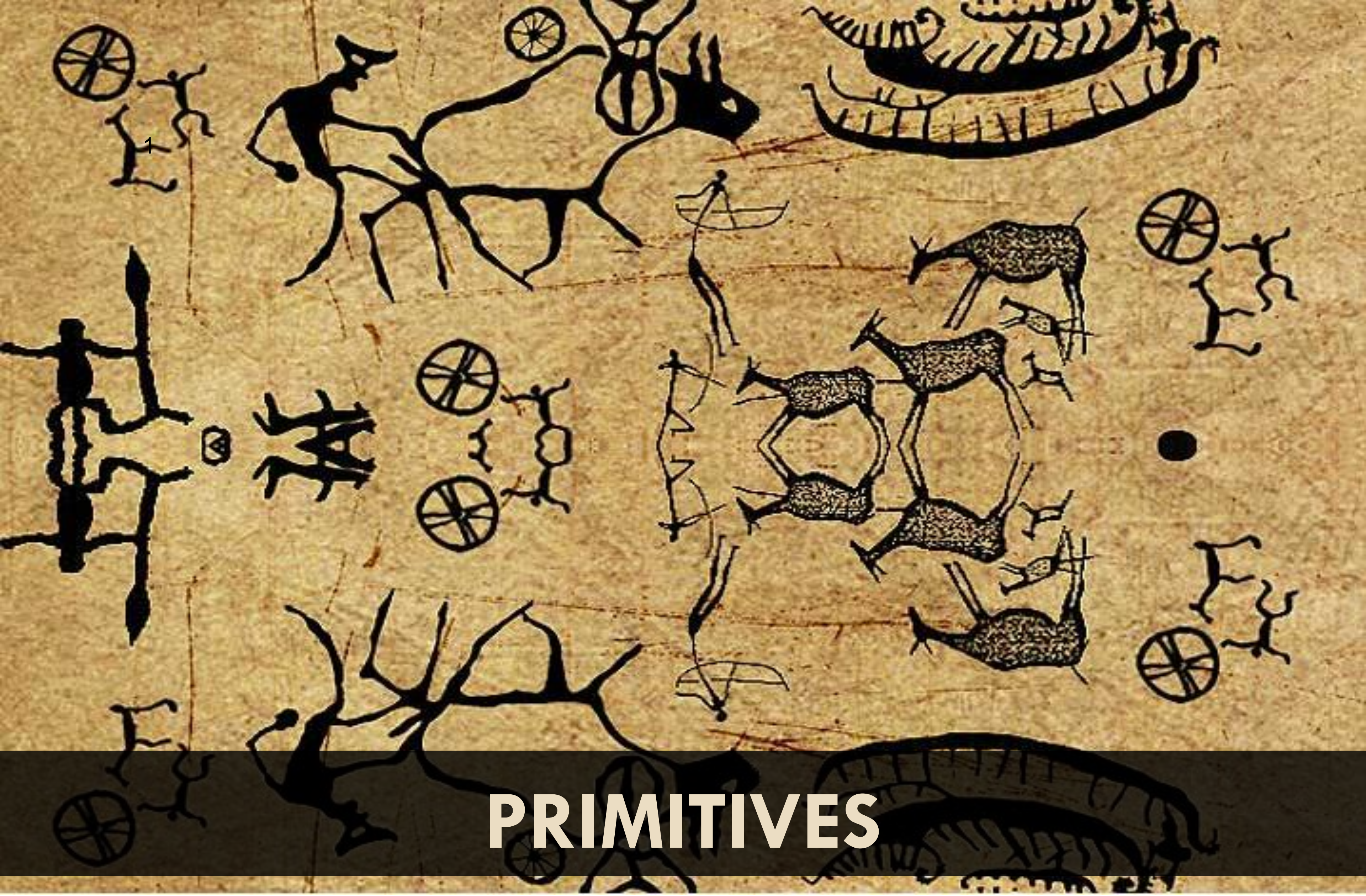


Camera Pixel Translation

$$\tan\frac{FovY}{2} = \frac{a}{c}$$




```
//parameter initialization
Double h = 2.0 * tan(FovY/2.0);
Double w = h * aspect;
Vector3 W = -(T - P).Normalized;
Vector3 U = (Up x W).Normalized;
Vector3 V = (W x U);
//ur, uc computation in for cycle
Double ur = h * (Double)r / (Double)Heigh - h / 2;
Double uc = w * (Double)c / (Double)Width - w / 2;
//ray setting
Vector3 rayDir = P + uc * U + ur * V - W;
rayDir = (rayDir - P).Normalized;
```



PRIMITIVES

Template 1

Computer Graphics 2

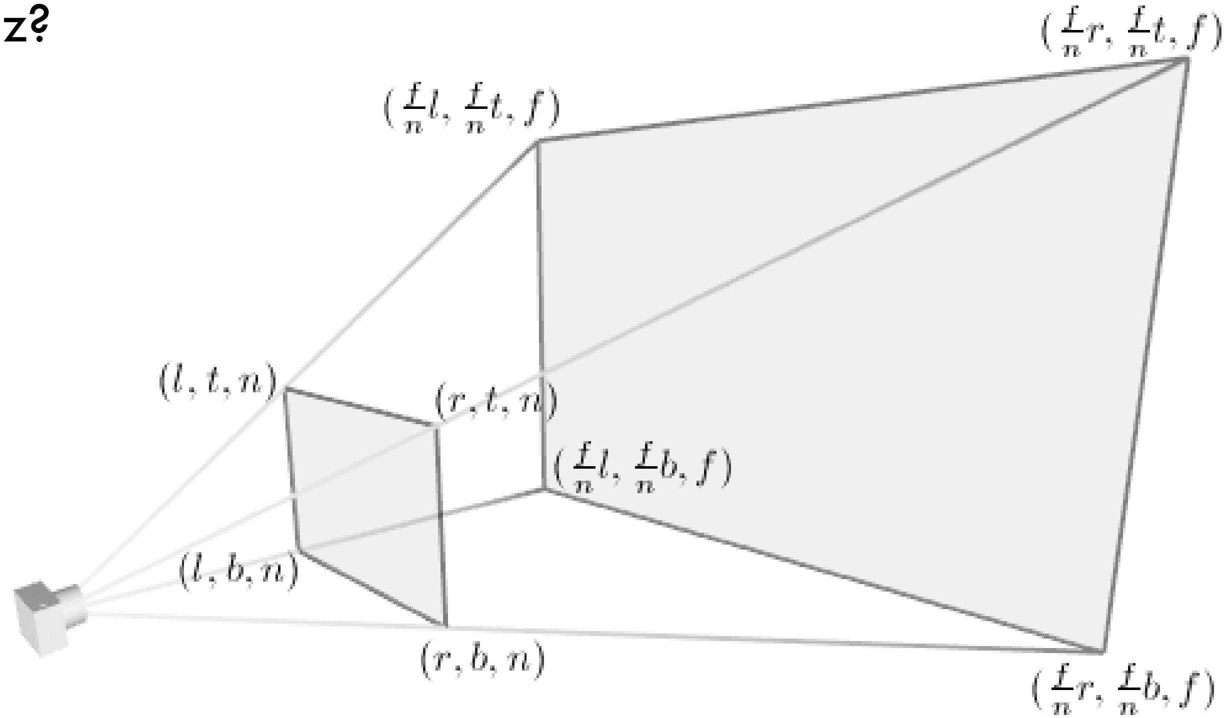
Primitive Shading

- Light source located at camera position
- Lower light intensity of distant objects
- Creates illusion of depth

```
Double attenuation = 1 - ray.HitParameter / (2 * (Target - Position)).Length;
return attenuation * ray.HitModel.Color;
```

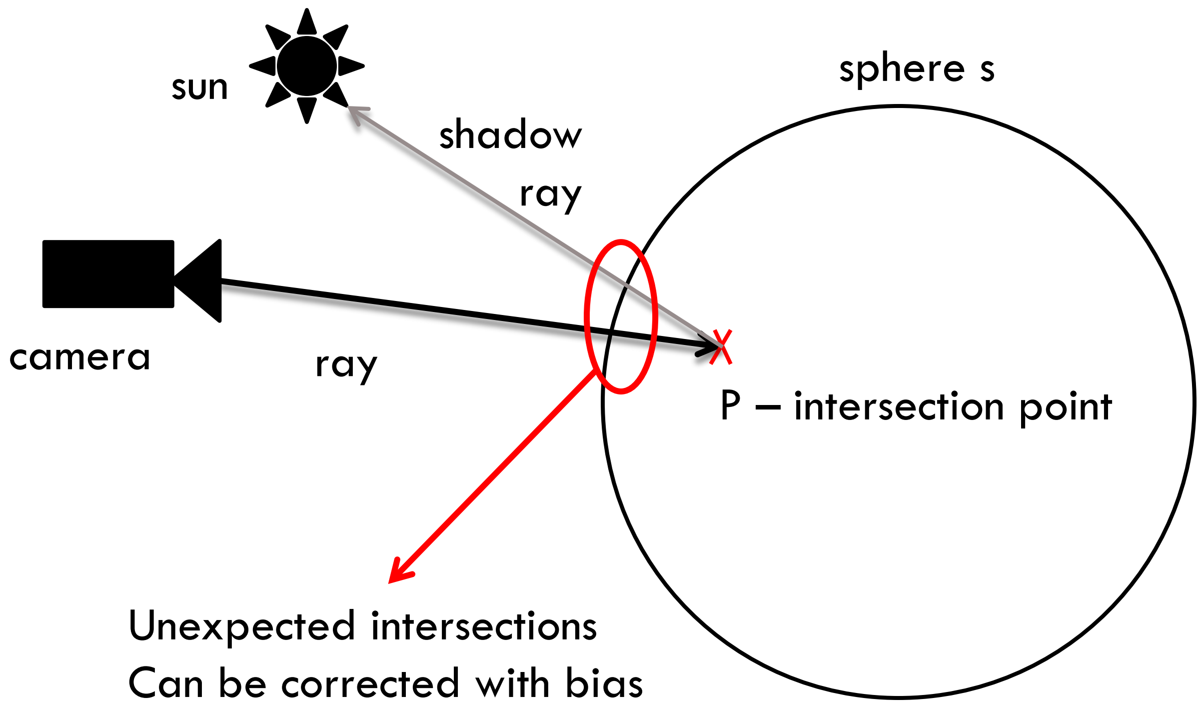
zNear & zFar

- Objects too close to camera would block all visible space
  - zNear clips objects too close
- Objects too far from camera are negligibly small
  - zFar clips invisible objects
- Why z?



Bias

- In computers: $Double \subset \mathbb{Q}$
- We use bias to correct for missing numbers
- Bias value depends on scene



AABB (Axis Aligned Bounding Box)

- Defined by two points representing minimum and maximum extend of the box B_0 and B_1
- Intersection parameter can be calculated for each axis aligned plane defining the AABB ($t_{0,x}$, $t_{1,x}$, $t_{0,y}$, $t_{1,y}$, $t_{0,z}$, $t_{1,z}$)

AABB – intersection parameters

For x coordinate:

$$r(t) = O + t\boldsymbol{r}$$

$$y = B_{0,x}$$

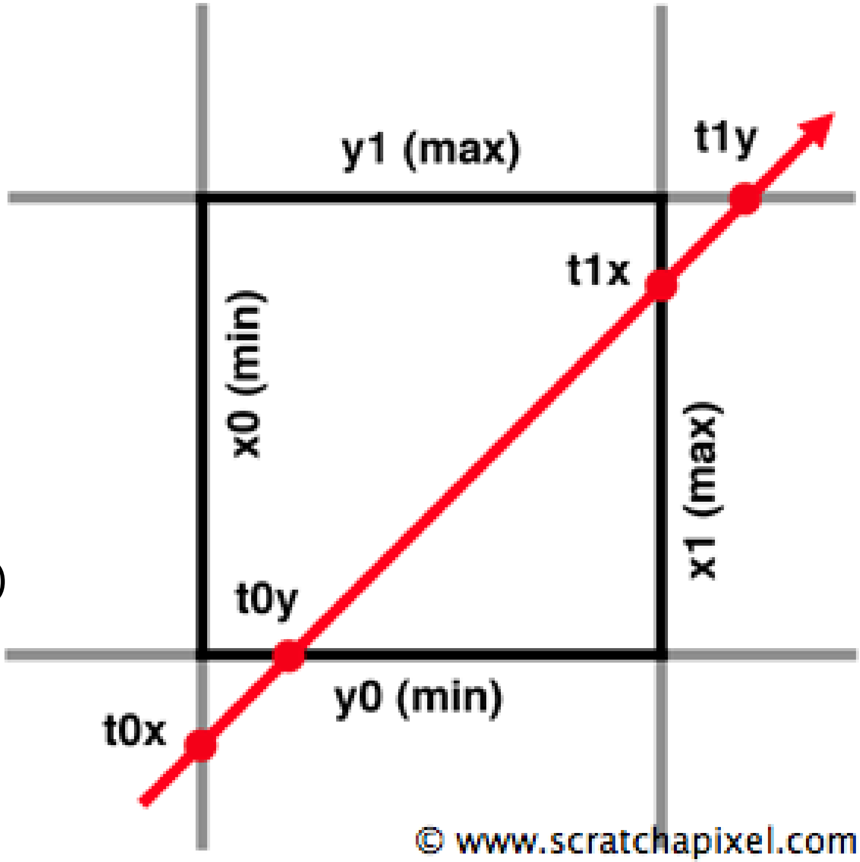
$$O_x + tr_x = B_{0,x}$$

$$t_{0,x} = \frac{B_{0,x} - O_x}{r_x}$$

2D case:

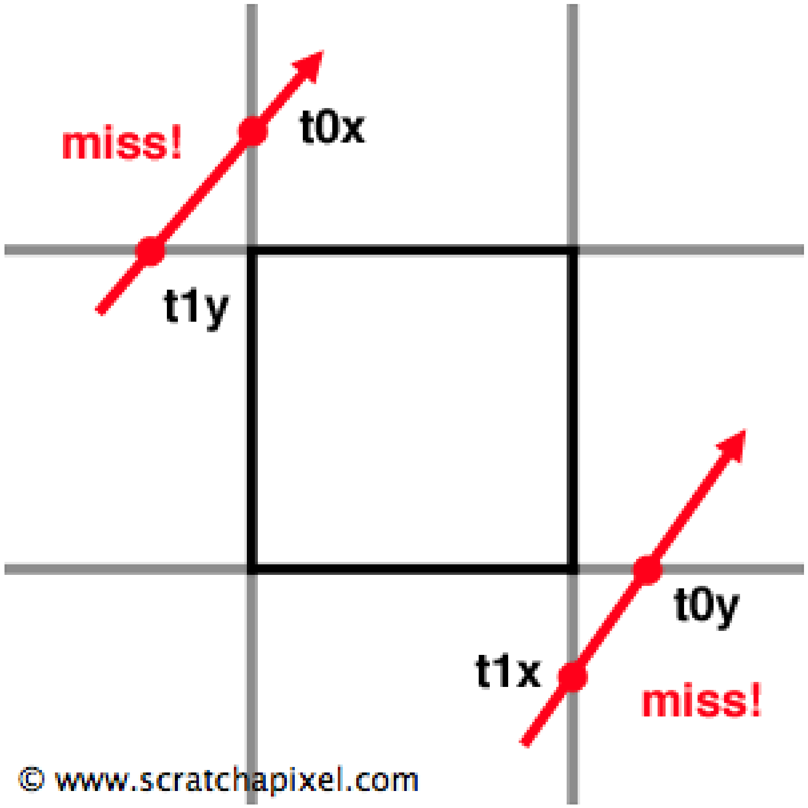
$$t_{min} = \max(\min(t_{0,x}, t_{1,x}), \min(t_{0,y}, t_{1,y}))$$

$$t_{max} = \min(\max(t_{0,x}, t_{1,x}), \max(t_{0,y}, t_{1,y}))$$



AABB checking for intersection

- Intersection actually occurs iff. $t_{min} \leq t_{max}$
- Resulting hit parameter is t_{min}



Sphere

$$\|X - C\|^2 - R^2 = 0$$

- Defined by center point C and radius R
- Intersection point can be solved analytically or geometrically

Sphere – Geometric Solution

$t_0 = t_{ca} - t_{hc}$ $\quad$ $t_1 = t_{ca} + t_{hc}$

$P = O + t_0\boldsymbol{r}$ $\quad$ $P' = O + t_1\boldsymbol{r}$

$\mathbf{L} = C - O$ $\quad$ $t_{ca} = \boldsymbol{L} \cdot \boldsymbol{r}$

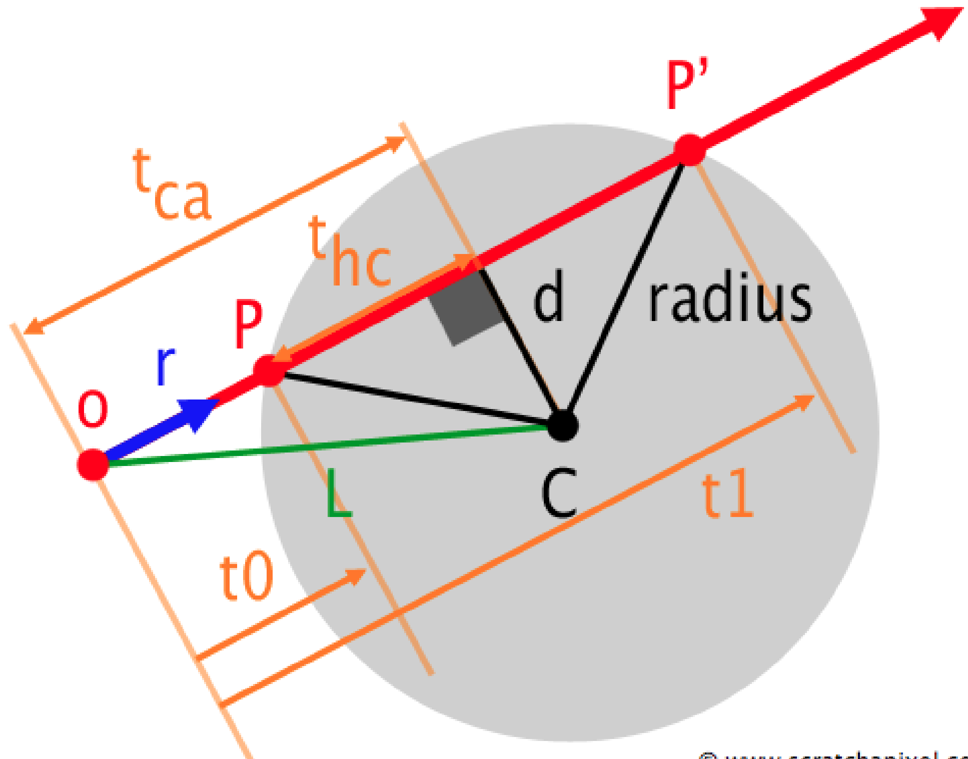
t_{ca} should be greater than zero.
What does $\boldsymbol{L} \cdot \boldsymbol{r}$ represent?
Using Pythagorean theorem:

$$d^2 + t_{ca}^2 = \boldsymbol{L}^2$$

$$d = \sqrt{\boldsymbol{L}^2 - t_{ca}^2}, 0 \leq d \leq R$$

$$d^2 + t_{hc}^2 = R^2$$

$$t_{hc}^2 = \sqrt{R^2 - d^2}$$



Sphere – Analytical Solution

$$\|X - C\|^2 - R^2 = 0$$

$$\|O + t\boldsymbol{r} - C\|^2 - R^2 = 0$$

$$t^2(\boldsymbol{r} \cdot \boldsymbol{r}) + 2t\big(\boldsymbol{r} \cdot (O - C)\big) + (O - C)^2 - R^2 = 0$$

$$t^2 + 2t\big(\boldsymbol{r} \cdot (O - C)\big) + (O - C)^2 - R^2 = 0$$

$$at^2 + bt + c = 0$$

where:

$$a = 1$$

$$b = 2\big(\boldsymbol{r} \cdot (O - C)\big)$$

$$c = (O - C)^2 - R^2$$

Circle

- Defined with origin C, normal **n** and radius R
- Same computation as ray-plane intersection
- After computing intersection parameter t we should check if $\|(O + t\boldsymbol{r}) - C\| \leq R$

Triangle

- Defined by three points A, B, C
- Intersection can be found using barycentric coordinates

$$\mathrm{P}(u, v) = (1 - u - v) * A + u * B + v * C$$

where: $\mathrm{u} > 0$

$\mathrm{v} > 0$

$\mathrm{u} + \mathrm{v} \leq 1$

If ray intersects triangle they have a common point:

$$O + t\boldsymbol{r} = (1 - u - v) * A + u * B + v * C$$

Questions?